An Advanced Operator Interface Design with Preview/Predictive Displays for Ground-Controlled Space Telerobotic Servicing

Won S. Kim, Paul S. Schenker, Antal K. Bejczy
Jet Propulsion laboratory
California institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91109

Stephen Leake and Stanford Ollendorf
NASA Goddard Space Flight Center
Code 714.1, Bldg. 11
Greenbelt, MD 20771

ABSTRACT

Recent advances in real-time 3-D graphics and graphical user interface (GUI) technologies enable development of advanced operator interfaces for telerobotic systems. In particular, we have employed Preview/predictive displays with calibrated graphics overlay and X/Motif-based GUI's for efficient and reliable ground-controlled space telerobotic servicing under communication time delay. High fidelity preview/predictive displays have been achieved by an operator-interactive camera calibration and object localization technique that enables reliable matching of simulated 3-D graphics models with the remote site task environment. The developed graphical operator interface supporting telerobotic operations with High-fidelity preview/predictive displays have been successfully utilized in demonstrating a ground-simulated ORU (orbital Replacement Unit) changeout remote servicing task by remotely operating a robot arm at NASA Goddard Space Flight Center from the Jet Propulsion laboratory under a varying time delay of up to several seconds. The positioning alignment accuracy achieved by this technique with four camera views was about $\pm 5mm$ for a tool insertion in the servicing task.

1. INTRODUCTION

Ground control of space robots has potential operational benefits in future space missions. Possible future applications include ground-controlled remote maintenance/repair servicing of spacecrafts including Space Station Freedom, ground-controlled telescience experiments, and ground-controlled remote assembly/construction work on the moon or Mars. An imminent potential application includes ground-controlled telerobotic servicing of the Hubble Space Telescope (HST) to assist EVA (extravehicular activity; space walk) astronauts performing a maintenance mission in the Space Shuttle cargo bay. The HST requires a periodic maintenance for every 5 to 1 years, for instance, to replace batteries, instruments, etc.. In a conceivable telerobot-assisted EVA maintenance scenario, EVA astronauts will capture and berth the HST on the Shuttle bay and perform critical tasks, while some other tasks such as open/close HST tool box, deploy/stow crew aids, and replace ORU's can be potentially carried out by telerobotic operations controlled from the ground. This telerobotic assistance is expected to reduce astronauts' EVA time, and save operational cost.

In such ground-controlled remote operations, however, there is an unavoidable communication time delay. When the existing NASA communication facilities are utilized, the round-trip time delay between the ground station and the low Earth orbit is expected to be 4 to 8 s to relay data via several communications satellites (e.g., TDRS and Domsat) and ground stations (e.g., White Sands ground station in New Mexico and Mission Control Center at Johnson Space Center). As the communication time delay increases beyond 0.5 s, it becomes more difficult for the human operator to perform remote

manipulation tasks, Two important and promising schemes for enhancing telemanipulation task performance under communication time delay are shared compliance control 9][and predictive display [2], [3], [8], [13], [15].

Wc recently developed an advanced operator interface that supports preview/predictive displays developed at Jet Propulsion laboratory and wrist force/torque ("wrench") sensor referenced compliance/impedance control implemented NASA Goddard Space night Center GSFC). Using the developed operator interface, in May 1993 we successfully demonstrated a JPL-God ard ORU changeout remote servicing task. In this demonstration, JPL acted as the operator site simulating the ground control station, and GSFC, more than 2,500 miles away from JPL, acted as the remote work site with a life-size satellite task mock-up. In this paper, we describe the operator interface design and its use in the recent JPL-Goddard demonstration.

2 PREVIEW/PREDICTIVE DISPLAYS

Noyes and Sheridan in the MIT Man-Machine Systems Laboratory [13] built the first predictive display for telemanipulation by using a stick-figure graphics model of the robot arm overlaid on the delayed video picture of the actual arm. In this predictive display, the operator drives the graphics model which responds immediately to the human operator's hand controller command, while the actual video image of the arm responds with time delay, thus following the graphics model. In effect, the graphics mode] leads or predicts the actual robot arm motion. The effectiveness of the predictive display technique was demonstrated through several experiments using simple models of the manipulator arm and simple tasks [15].

In our design, stick-figure-type predictive display technology has been extended to high-fidelity 3-I) predictive display technology for applications to ground-controlled telerobotic servicing in space with communication time delay. High fidelity is achieved by 1) precise 3-D graphics modeling/rendering, and 2) operator-interactive reliable camera calibration and object localization that enable calibrated overlay of graphics models on the live video of quasi-static telerobotic task environments, In contrast to previous predictive displays in which only the robot arm graphics model is overlaid on live video through a camera calibration procedure, in our design both the robot arm and the object graphics models are overlaid through an additional object localization procedure. Although various camera calibration and object, localization algorithms [I], [5], [18] have been reported, in our design we use newly developed operator-interactive camera calibration [7] and object localization [10] algorithms to achieve reliable, accurate matching of graphics models with actual camera views of the remote site task environment.

in the original predictive display, the operator-commanded hand controller motion drives both the simulated graphics model (without delay) and the real robot arm (with communication time delay) simultaneously, In our design, we adopted a new strategy by combining preview and predictive displays to enhance safety and reliability in remote servicing operations such as in ground-controlled telerobotic servicing in space. In this preview/predictive display strategy, the operator first interacts with the graphically simulated "virtual environment" by driving the simulated graphics model against the background remote-site video camera view. The operator then previews and verifies the simulated robot motion. Only after the preview verification, the operator sends the motion command to the remote site for actual motion execution. This sequential (as opposed to simultaneous) preview/execution operation is repeated for each new task segment.

Preview/predictive displays can be used both in the teleoperation mode and in the computer-generated trajectory mode. A typical scenario to perform a task segment with preview/predictive displays is as follows. 1) The operator generates the robot arm trajectory either by driving the simulated robot arm with a hand controller (in the teleoperation mode) or by selecting the target frame (tag point) and requesting the system to generate the straight line trajectory (in the computer-generated

trajectory mode), and the simulated robot motion trajectory is recorded. 2) The operator plays back the recorded robot motion with an appropriate time scale by again driving the simulated arm to preview and verify the robot motion trajectory. This preview verification is important to ensure operational safety. 3) The operator sends the verified trajectory to the remote site, and the remote system stores the trajectory data in a buffer. This trajectory data buffering ensures accurate motion execution even with slow or abrupt change in the communication time delay. 4) After the receipt of the whole trajectory, the remote system executes the robot motion trajectory command to drive the actual robot arm. During the execution, force/torque sensor referenced compliance/impedance control can be activated. In the local site, the operator monitors the command execution by visually observing the preview/ predictive display updated with the returned video image of the robot arm motion. 5) After the completion of the robot motion execution, the graphics model of the arm is updated with the actual final robot joint angles. This update procedure not only eliminates accumulation of motion execution errors but also enables the preview/predictive display to be useful even when the compliance/impedance control is activated in the remote site, for example, during the performance of a contact or insertion task.

In our graphical operator interface design, two Silicon Graphics workstations and one NTSC video monitor are currently used. The primary workstation (RIS-4D 3/10 VGX) is used for preview/predictive displays and for various GUI's. A Silicon Graphics VideoLab board installed in the primary workstation captures the live video picture at 30 frames/s, and supports real-time graphics overlay to appear both on the high-resolution (1280X10 24) workstation monitor and on the low-resolution video monitor simultaneously. The second workstation (1}{1S-41)/70 GT) is solely used for sensor data display, providing graphical visualization of robot arm joint angles, 6-dof force/torque sensor data ("wrench vector"), and capaciflector proximity sensor data [4].

The operator interface software [6], [11] was all written in C using X, Motif, Wcl, GL, and GLX. The X window system [9] is an industry standard software enabling development of "device-independent" portable GUI's. 1' ie Motif widget set is an industry-prevailing X toolkit that provides useful widgets such as scroll bars, menus, and buttons, The Wcl Widget Creation Library [16] enables X resource files to specify a widget hierarchy (parent-child relationship tree), widget types, and bindings of callbacks. Since widget creations can be conveniently defined in extended X resource files, the use of Wcl can greatly reduce the programming effort. GL is the Silicon Graphics standard graphics library for graphics rendering. GLX allows to create a special window that accepts both X and GL functions for the Silicon Graphics workstation.

3 CAMERA CALIBRATION

Before starting actual telerobotic task performance with preview/predictive displays, the operator must perform the camera calibration and object localization procedures to enable calibrated graphics overlay necessary for preview/predictive displays. In our camera calibration procedure, the robot arm itself is used as a calibration fixture, eliminating a difficult and error-prone procedure of measuring the precise position and orientation of a separate calibration fixture relative to the robot base. The operator enters the correspondence information between 3-D object model points of the robot arm and 2-D camera image points by using a mouse, In order to improve the calibration accuracy over the normal operating region of the robot arm, the operator enters corresponding points data for several different poses of the robot arm. Thereafter the system computes the camera calibration matrix. This calibration procedure is repeated for each of the camera views desired.

Fig. 1 shows the graphical operator interface used during the operator-interactive camera calibration. As the operator selects the "camera calibration" button from the main menu bar of the lower right window, the camera calibration GUI pops up on the upper right window, The solid-shaded 3-D graphics is displayed on the upper left window, and the live (or stored) video picture received from the remote site appears on the lower left window. At this stage the graphics view is in general not aligned with the camera view. In fact, the operator is allowed to change the graphics viewing condition (view angle,

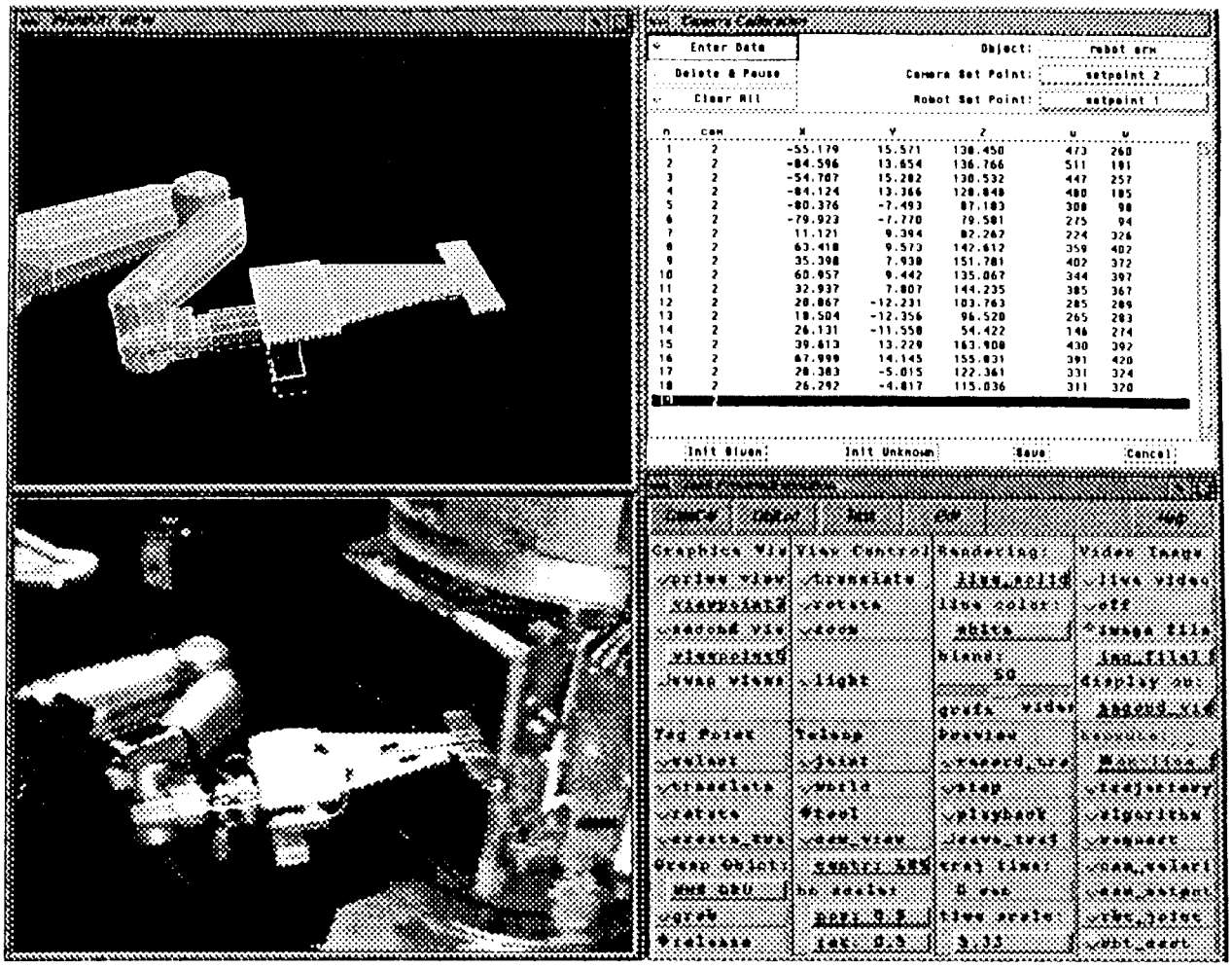


Figure 1: Graphical operator interface during camera calibration.

position, and zoom) at any time during this data entry mode as desired. To enter corresponding point data, the operator first picks an object model point of the robot arm graphics with a mouse (upper left window), and then clicks the corresponding image point on the video image of the robot arm (lower left window). When an object point is picked by a mouse, the system actually obtains the 3-I) position of the object point by utilizing 1) the CAD (computer-aided design) geometric model of the robot arm graphics, 2) the actual joint angle values of the robot arm on the video picture, and 3) the "pick" function call of the Silicon Graphics standard GL graphics library, As the operator clicks 3-D object model points or 2-D image points, their coordinate values appear on the scrolled list widget of the camera calibration GUI. The operator can use the "robot set point" button to move the remote robot arm in a different pose and enter more corresponding points. The operator can also delete and re-enter corresponding points data. When all desired object points and their corresponding image points at different arm poses are entered, the operator can request the system to compute the camera calibration parameters.

The computation of the camera calibration parameters is based on the ideal pinhole camera model. Under this assumption, the image formation of the camera is described by a linear perspective projection, and the relation between 3-I) graphics model points and the corresponding 2-D video camera image points can be described by a 4X3 calibration matrix. Both linear and nonlinear least-squares algorithms

[7] are used in camera calibra ion computations.

The operator is provided with two options: 1) initial unknown and 2) initial given. When an approximate solution is not available, the operator chooses the "initial unknown" button to request the system to proceed with the following two-step procedure for efficient computation: (i) first, obtain an

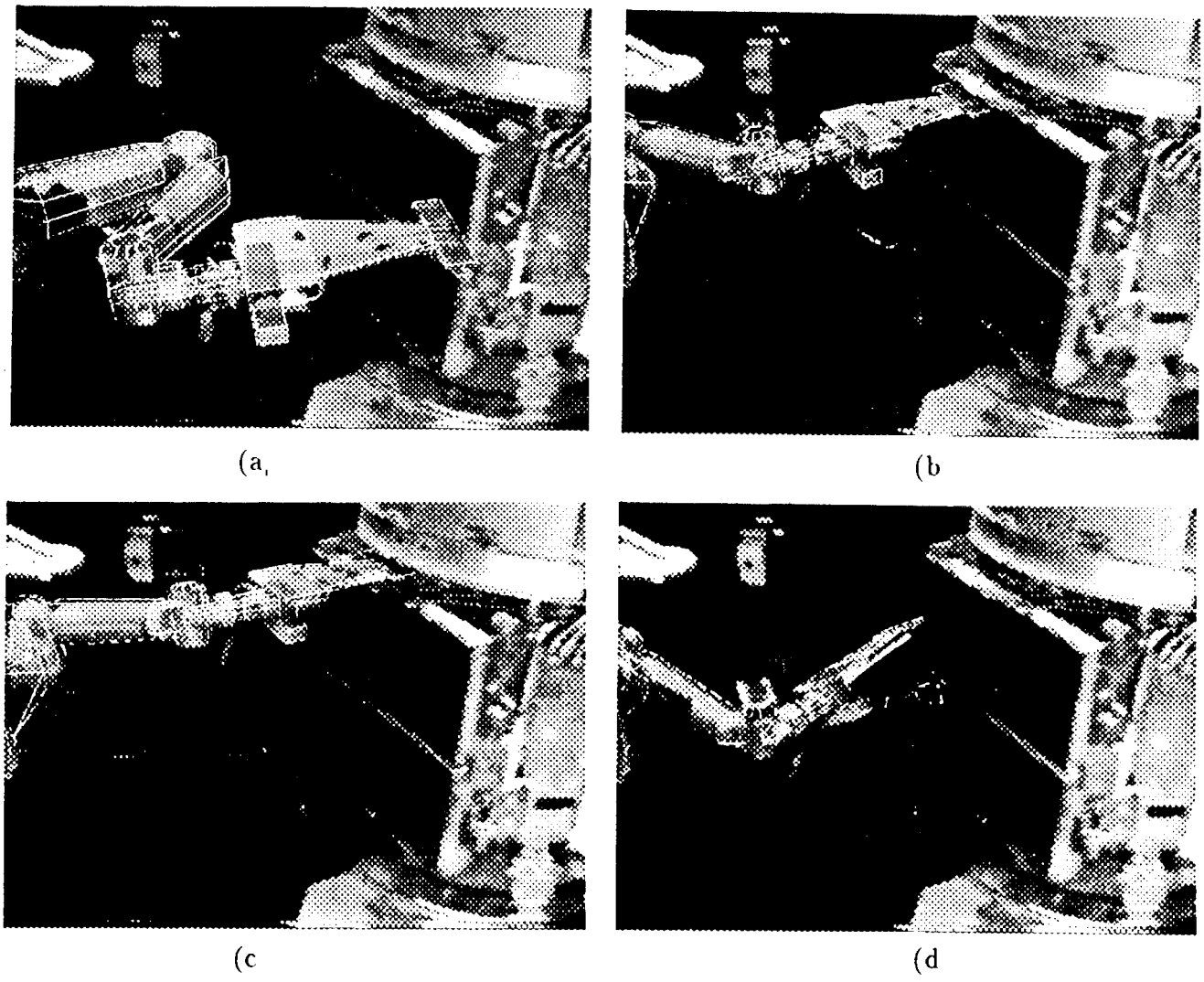


Figure 2: Calibrated overlays of the robot arm graphics model on the live video picture after the camera calibration at four different arm poses.

approximate solution by the linear algorithm, and then (ii) apply the nonlinear algorithm by using the linear least-squares solution as an initial guess. When the number of corresponding points entered is only 4 or 5, or when all the object points lie on the same plane, the linear algorithm cannot be used, and the system must start with the nonlinear algorithm directly. Once the camera calibration matrix is obtained, the graphics model of the robot arm can be overlaid on the video camera view.

When an initial approximate solution. is known (for example, by the operator's approximate matching of the graphics model to the camera view or by a prior graphics simulation with task analysis and planning), the operator chooses the "initial given" button to request the system to start with the nonlinear algorithm directly. When the algorithm converges to a solution with the average error less than 5% between the 3-D object points projected on the image plane and the actual 2-D image points, the system assumes that the nonlinear least-squares solution is found. Otherwise, a new initial condition is tried by incrementing one of the rotation angles by 300 The 30° increment for all three rotation angles is repeated until a good solution with the average error less than 5% on the image plane is found.

Four camera views were calibrated for the JPL-Goddard remote servicing demonstration (see Section 6). In each camera calibration, the operator typically entered about 15 to 30 data points in total from 3 or 4 different arm poses. For the side-view and oblique-view cameras, the vertical field-of-view angles were both approximately $fovy = 32°$, and the average calibration errors between the projections of 3-D object points on the image plane and the actual 2-D image points were typically in the range of 0.5-0.9% (with 1.5-3.5% maximum errors). The object (robot arm) distance from these cameras was about 3 m, and the 0.7% average error on the image plane corresponds to 1.2 cm displacement error on the hypothetical plane 3 m in front of the camera. Calibrated graphics overlay examples for the oblique-view camera are shown in Fig. 2 at the four different arm poses used for the calibration. Two zoom settings were used for the overhead (front-view) camera, which was about 1 m away from the robot end effecter. For the wide-angle view ($fovy = 370$), the average error on the image plane was typically 0.5-0.9% (1.3-2.9% maximum error), and the 0.7% average error corresponds to 0.5 cm displacement error on the plane 1 m in front of the camera. For the zoom-in view ($fovy = 110$), the average error on the image plane was typically 1.3- 1.7'%0 (3.6-4.6% maximum error), and the 1.5% average error corresponds to 0.3 cm displacement error on the plane 1 m in front of the camera.

4 OBJECT LOCALIZATION

'] 'he above camera calibration procedure allows to overlay only the graphics model of the arm on the video camera view. A key novel feature of our new interface design is that it enables graphics overlay of objects as well as the robot arm by providing an operator- interactive object localization procedure that determines the object pose (position and orientation).

The operator's interactive data entry procedure and its operator interface for the object localization procedure are essentially identical to those for camera calibration (See Fig. 1 and Section 3), except that the operator enters corresponding data points for an object (not the robot arm this time) with several different camera views. The operator uses the "camera set point" button to select a camera and define its set point for remotely controllable pan, tilt, and zoom parameters. There are several algorithms available in the literature to determine the 3-D object pose from a given 2-I) camera view. In our computation, a projection-based linear/nonlinear algorithm extended to allow object localization for any number of multiple camera views [10] is used. Again, the operator is provided with two computing procedure options: 1) initial unknown and 2) initial given. Computing procedures for these two options are essentially the same as those in camera calibration described in Section 3.

Once the camera calibration and object localization are completed, the graphics models of both the robot arm and the object can be overlaid with high fidelity on the corresponding actual video images in a given video camera view. The arm and object graphics models can be overlaid in a wire-frame or in a solid-shaded polygonal rendering, with varying levels of transparency, providing different visual effects

to the operator for different task details. The hidden lines can be removed or retained by the operator, depending upon the information needs in a given task. The above object localization method was applied to locate an ORU (Orbital Replacement Unit) in the JPL-Goddard remote servicing demonstration (see Section 6) by using the four calibrated camera views described in Section 3. Calibrated graphics overlay examples after the camera calibration and object localization are shown in Fig. 3 for the four camera views.

The average position error (with standard deviation in parenthesis) of the object localization of the ORU from 10 measurements was -0.53 *cm* (0.11 cm), 0.53 *cm* (0.17 *cm*), and 1.4 *cm* (0.20 *cm*) for the horizontal, vertical, and insertion axes of the ORU hole, respectively. The average orientation error (with standard deviation in parenthesis) was -0.35° (0.170), 0.36° (0.190), and -1.1° (0.24°) about the horizontal, vertical, and insertion axes, respectively. Note that the positioning alignment accuracy achieved by our technique with four camera views was about ±5 *mm* for inserting a tool into the ORU hole.

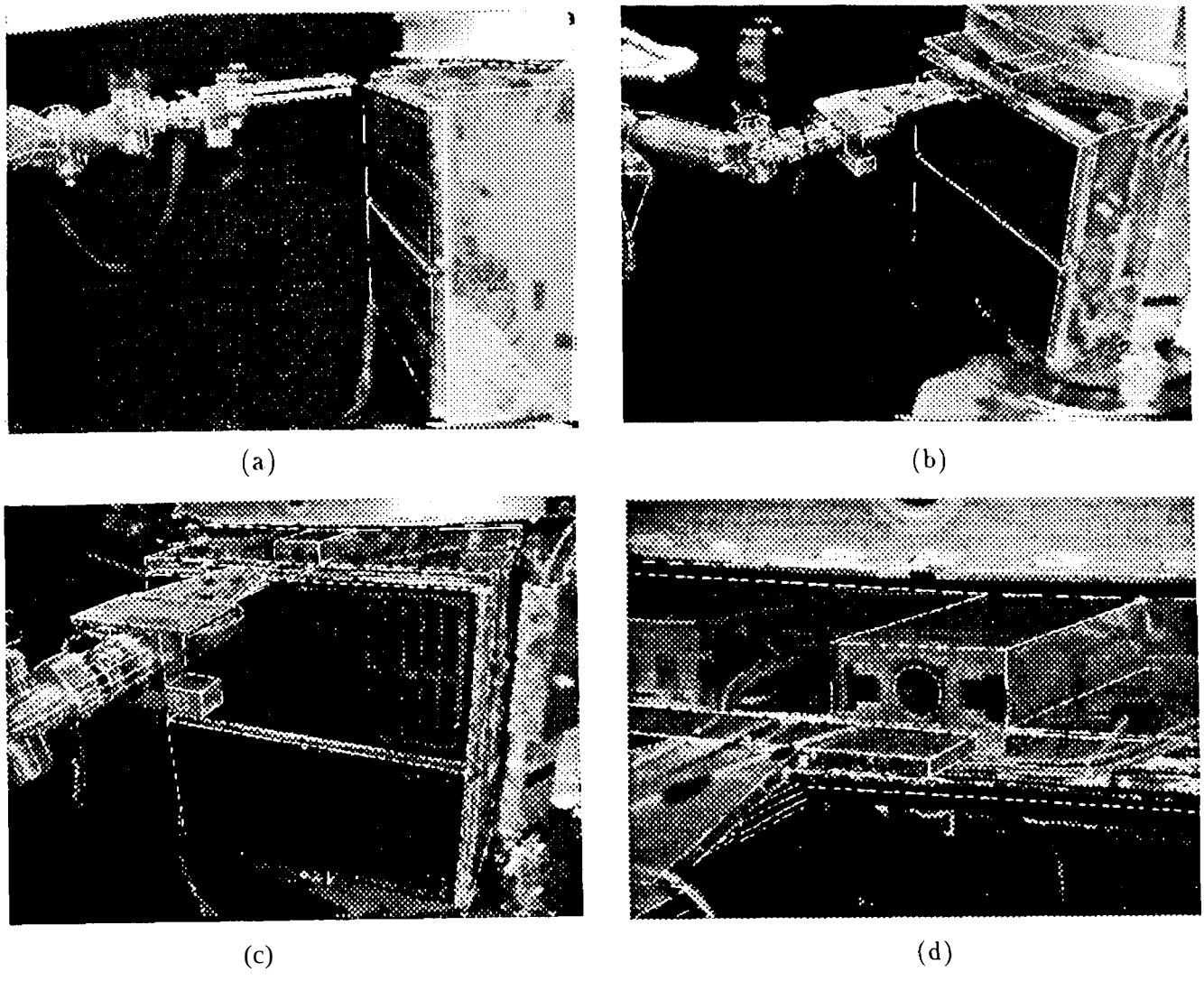
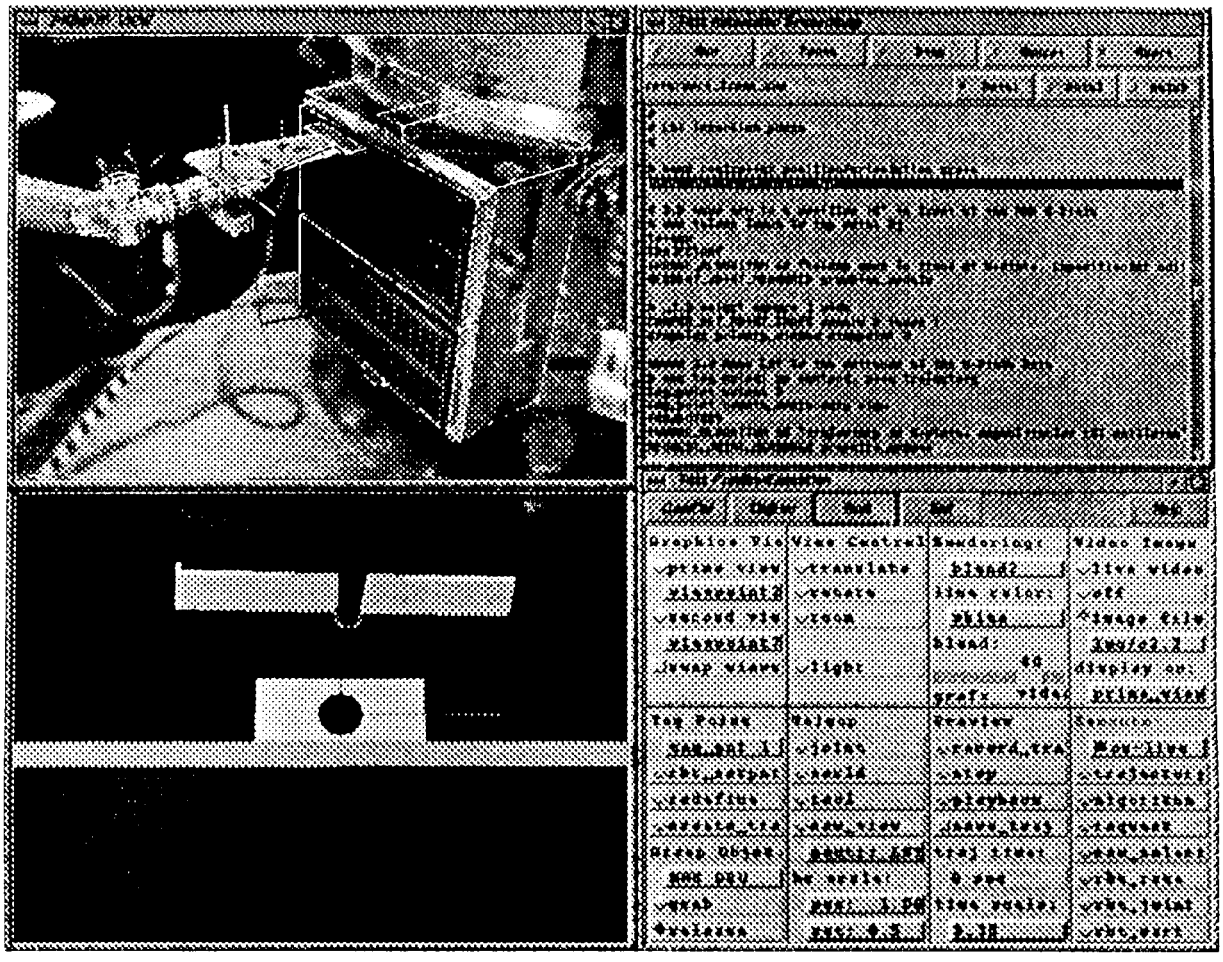


Figure 3: Calibrated overlays of both robot arm and ORU graphics models on the live video picture after the camera calibration and object localization with four camera views. (a) side-view camera, (b) oblique-view camera, (c) overhead camera with wide angle, and (d) overhead camera with zoom in.

5 TASK EXECUTION

A top-level screen layout on the primary workstation used during, the actual task execution is shown in Fig. 4. It consists of two NTSC-resolution (646x486) windows on the left side and two slightly smaller windows on the right side. A calibrated preview/predictive graphics overlay on the live video picture appears on the upper left window, and it also appears on the full screen of the 19 inch NTSC monitor for better viewing. A 3-I) graphics display of either a calibrated view that matches with one of the camera views or an operator-defined synthetic (virtual) camera view (of any desired viewing angle, position and zoom) appears on the lower left window. A task auto sequencing GUI appears on the upper right window, and the graphics/robot control main GUI on the lower right window.



'igure 4: Graphical operator interface during the task execution with preview/predictive displays

After the completion of the camera calibration and object localization procedures, the operator can ac ually perform a remote servicing task with preview/predictive displays. The main GUI consists of a menu bar and 8 panels, The menu bar is used to pop up a new GUI on the upper right window (for example, object localization GUI). The upper 4 panels provide an interface for graphics control: 1) first panel for graphics view selection among calibrated graphics views or operator-defined virtual camera views, 2) second panel for graphics view control with a mouse such as graphics translation, rotation, and zoom, 3) third panel for graphics rendering model selection (wire-frame, solid-shaded, wire-frame with hidden line removal, wire-frame with semi-transparent solid surface model), and the last panel for

video image selection (no video, live video image, or stored video image file).

The lower 4 panels of the main GUI (Fig. 9) provide an interface for robot control. The first panel allows the operator to use tag points (target frames or via points) to define robot motion trajectory. To generate a trajectory, the operator just designates a target position, and then the system automatically generates the robot motion trajectory. Note that in this coil-l~l\lter-~e]lerated trajectory mode, the operator does not use a hand controller. The first panel also has the object grab/release option to indicate the grasp status to the graphics simulation. When an object (e.g., an ORU in our demonstration) is grasped by a simulated robot arm, the grasped object moves together with the simulated arm.

The second panel allows the operator to select a desired cartesian control mode for the hand controller control: world (robot base), tool (end effecter), and camera-view referenced control. The operator is also allowed to define the origin (center of rotation) of the cartesian control reference frame, for example, end effecter, tool tip, or an object in grasp. The operator can also set the position and orientation gains of the hand controller motion, The same inverse kinematic or inverse Jacobian cartesian control software that, drives the actual remote robot arm is used to drive the local-site simulated robot arm [12], [14].

The third panel allows the operator to record the hand controller motion, stop recording, and play back the recorded motion trajectory for preview simulation. A Multibus-I based real-time system reads the hand controller motion data and sends the data to the UN IX-based Silicon Graphics workstation at 30 Hz through a 9600-baud serial 1/0 line. 'I'he serial I/O buffer and trajectory data buffer queues were effectively used to avoid any missing data [1 7]. The time scale concept is employed to allow the operator to change the speed in converting the hand controller motion trajectory to the robot arm motion trajectory. Time scaling appears to be particularly useful when the remote arm is very slow.

The last panel allows the operator to send an execution command to the remote site interactively. The execution commands currently supported include robot arm motion trajectory (TRAJECTORY), control algorithm selection (INVOKE_ALGORITHM), sensor data request such as joint angles, cartesian pose, force/torque sensor data ("wrench vector"), and capaciflector readings (REQUEST), camera selection and set point (CAMERA), robot joint move (GOTO_JOINT), and robot cartesian move commands (GOTO_CART).

The task auto sequence GUI displays a selected auto sequence script on the scrolled list window, and the current command to be executed is highlighted. The operator can execute the highlighted command by clicking the "step" button. The operator can interrupt the current execution by "cancel" button, or abort the remaining script completely by clicking the "abort" button. Two types of commands exist: local and remote execution commands. Local execution commands that effect only the local site include Graphics, Video, Camcal, Objloc, Reference_frame, Tag-point, and Object grab /relase commands. All the remote execution commands supported by the last panel of the main GUI are also supported by the task auto sequence GUI.

6 ORU CHANGEOUT REMOTE SERVICING DEMONSTRATION

The developed operator interface described above were successfully utilized in demonstrating a ground-simulated ORU thangeout remote servicing task. The demonstration was to show potential capabilities of ground- contro led telerobotic servicing by remotely operating a robot arm at NASA Goddard Space Flight Center from the Jet Propulsion Laboratory. The demonstration task performed was to remove an old MMS (Multi-Mission Servicing) ORU module from the Explorer Platform (EP) spacecraft mockup and install a new one (the same ORU in our demo) by using a Robotics Research Cor-

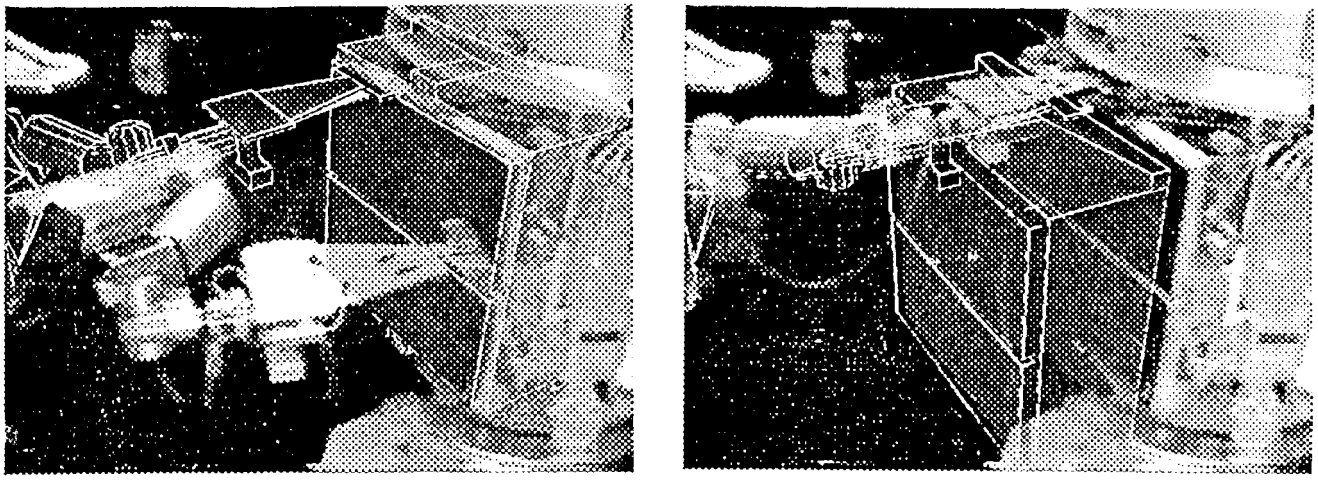
poration K-1607 robot arm and a Lightweight Servicing Tool (LST; socket driver power tool) mounted at the end of the arm [12]. The EP spacecraft, which was launched in 1992, is a modular mission spacecraft, carrying several modules that can be replaced on orbit by astronauts.

In our demonstration, the NASA Select NTSC Television broadcasting channel was used (30 frames/s) to transmit the live video image from NASA-GSFC to JPL. A 'I'CI'/II' socket communication with ethernet connection through the Internet computer network was used for a bidirectional command/data link. The round-trip Internet socket communication delay between JPL and NASA-GSFC was measured about 0.1 s on the average, although there were often long time delays (e. g., a 10-minute testing indicated that about 0.8% of the delays was longer than 0.5 s and about 0.01% was longer than 4 s).

The ORU changeout task scenario used in the remote servicing demonstration had the following sequence. 1) Perform camera calibration. 2) Perform object localization to determine the ORU pose, 3) Move the arm from the starting position to a position where the LST tip is about 20 cm in front of the entrance of the hole on the ORU module. 4) Move the LST to the immediate entrance of the hole. 5) insert the 1, S'1'. 6) Latch the LST to the ORU. 7) Turn on the power tool to loosen the screw. 8) Pull out the ORU by 5 cm. 9) Continue to withdraw the ORU so that it is about 15 cm apart from the satellite. 10) Move the ORU to a stow position. 11) Move the ORU back to 15 cm in front of the satellite frame. 12) Align the ORU for insertion. 13) Insert the ORU. 14) Turn on power tool to tighten the screw. 15) Unlatch the LST from the ORU. 16) Pull out the 1ST to about 20 cm away from the ORU. 17) Finally, move the arm back to the starting position.

Steps 5, 6, 7, 8, 13, 14, and 15 were executed autonomously by invoking an appropriate algorithm using the INVOKE_ALGORITH M command. During these steps the robot arm motion involves actual contact with the task environment for tool or module insertion or removal actions, and thus were aided by wrist force/torque ("wrench") sensor referenced automatic compliance/impedance cent rol implemented at GSFC. Steps 3, 4, 9, 10, 11, 12, 16, and 17 were executed by a TRAJECTORY command, where the robot arm trajectory data was generated either by teleoperation using a hand controller or by a computer with an operator designated target point. It is important to note that several fixed target frames (tag points) were pre-defined relative to the ORU to facilitate generation of trajectories. Through tile object localization (Step 2), the accurate ORU pose was determined, and so were the accurate poses of these target frames fixed relative to the ORU. In the teleoperation mode, these target frames were merely used as a visual aid to the operator in generating a robot arm trajectory with a hand controller. In the computer-generated trajectory mode, the operator just designated the target frame, and then the computer generated a clean straight line trajectory from the current robot arm pose to the designated frame". The teleoperation mode using a hand controller was helpful for fine alignment to compensate for any errors caused by imperfect modeling and gravity compensation, while the computer-generated trajectory mode was very helpful for global motion.

High-fidelity predictive/preview displays were very useful for the operator to generate a robot arm trajectory with confidence under communication time delay. 'I'he operator generated the overlaid robot graphics image motions by a hand controller or by computer control algorithms. Then the operator visually verified the correctness of the generated robot motions through previewing the simulated robot graphics image motions embedded into the monitor of an actual TV camera image of the work scene. Once verified, the recorded motion command was sent over to the GSFC robot control system. In order to eliminate the problem associated with the varying time delay in data transfer, the robot motion trajectory command was not executed at the GSFC control system until all the data blocks for the trajectory were received. A few seconds after the motion commands were transmitted to GSFC from JPL, the JPL operator could view the motion of the real arm on the same screen where the graphics arm image motion was previewed. When no contact was involved, the video image of the real arm basically followed the same trajectory, and stopped at the same position where the graphics arm image stopped earlier, When contact was involved, however, the final positions of the simulated graphics and actual robot arms could be quite different. For this reason, after the completion of the robot arm trajectory command, the simulated graphics arm was updated with the actual final robot joint angle values. This update eliminates accumulation of minor motion execution errors as well as large compensation errors due to the compliance/impedance control. Examples of preview/predictive displays with calibrated



(a) *(b)*

Figure 5: Preview/predictive displays with calibrated graphics overlay for the JPL-Goddard remote servicing demonstration of an ORU changeout task. (a) Step 3: approach the arm from the starting position to tile ORU to prepare for insertion, and (b) Step 9: continue to pull out the ORU.

graphics overlay during the performance of steps 3 and 9 are shown in Fig. 5.

7 CONCLUSION AND FUTURE PLANS

An advanced opcr'titer interface supporting telerobotic operations with high-fidelity preview/predictive displays were developed for applications to ground-controlled telerobotic servicing in space. High fidelity preview/predictive displays were achieved by using operator-interactive camera calibration and object localization methods. The developed methods enable reliable, accurate matching of graphics models to the remote site task environment with typically less than 2% average error 011 the image plane. The developed operator interface design was successfully utilized in the recent JPL-Goddard ground-simulated demonstration of an ORU changeout remote servicing task, showing the practical utility of IIi,p;h-fidelity predictive/preview display techniques combined with compliance control. The same techniques also have a wide range of terrestrial application possibilities. Future planned work includes: 1) simulated tests on other space application tasks like Hubble Space Telescope Servicing, and 2) interactive model building and intermittent model matching updates using model-based image processing.

8 ACKNOWLEDGMENT

This work was performed at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration. The authors would like to thank 11. Das, E. Paljug, and E. Barlow of JPL, and D. Henry and E. Cheung of GSFC for their contributions to make the demonstration successful.

9 REFERENCES

[]] D. H. Ballard and C. M. Brown, *Computer* Vision, Englewood Cliffs, NJ, Prentice-Hall, 1982.

[2] A. K. Bejczy and W. S. Kim, "Predictive Displays and Shared Compliance Control for Time-Delayed Telemanipulation," Proc. IEEE Int. Workshop on Intelligent Robots and Systems (IROS '90), pp. 407-412, Tsuchiura, Japan, July 1990

[3] A. K. Bejczy, and W. S. Kim, and S. Venema, "The Phantom Robot: Predictive Displays for Teleoperation with Time Delay," IEEE Int. Conf. on Robotics and Automation, pp. 546-551, Cincinnati, 011, May 1990.

[4] 11. Das, "Capaciflector Data Display," Jet Propulsion Laboratory Internal Document Interoffice Memorandum, 3474-92-092, Dec. 1992.

[5] B. K. P. Horn, Robot Vision. Cambridge, MA: MIT Press, 1986.

[6] W. S. Kim, "Graphical Operator Interface for Space Telerobotics," IEEE Int. Conf. on Robotics and Automation, Atlanta, GA, May 1993.

[7] W. S. Kim, "Graphics Overlay and Camera Calibration for Predictive Displays," Jet Propulsion Laboratory internal Document Engineering Memorandum 347-89-273, Dec. 1989.

[8] W. S. Kim and A. K. Bejczy, "Graphics Displays for Operator Aid in Telemanipulation," Proc. IEEE Conf. on Systems, Man, and Cybernetics, pp. 1059-1067, Charlottesville, VA, Oct. 1991.

[9] W. S. Kim, B. Hannaford, and A. K. Bejczy, "Force-Reflection and Shared Compliant Control in Operating Telemanipulators with Time Delay," IEEE Trans. on Robotics and Automation, vol. 8, 110.2, pp. 176-185, 1992

[10] W. S. Kim and L. Stark, "Cooperative Control of Visual Displays for Telemanipulation" IEEE Int. Conf. on Robotics and Automation, pp. 1327-1332, Scottsdale, AZ, 1989.

[11] W. S. Kim. K. S. 'I'so, and S. Hayati, "An Operator Interface Design for a Telerobotic Inspection System, " AIAA Aerospace Design Conference, AIAA 93-1160, Irvine, CA, Feb. 1993.

[12] S. Leake, "A Cartesian Force Reflecting Teleoperation System ," Int. J. of Computers and Electrical Engineering, vol. 17, no. 3, pp. 133-146, 1991.

[13] M. V. Noyes and T. B. Sheridan, "A Novel Predictor for Telemanipulation Through a Time Delay," Proc. of 20th Annual Conference on Manual Control, NASA Ames Research Center, Moffett Field, CA, 1984.

[14] E. Paljug, "Robot Software for the GSFC Teleoperation Demonstration, " Jet Propulsion Laboratory Internal Document Interoffice Memorandum, 3474-93-014, Feb. 1993.

[15] T. B. Sheridan, *Telerobotics, Automation, and Human* Supervisory *Control*, The MIT Press, Cambridge, MA, 1992.

[]6] D. E. Smyth, "WC] - Widget Creation Library, Mri - Motif Resource Interpreter, Ari -- Athena Resource Interpreter: An Easier Way to Develop Applications using Motif and Athena Widgets," Proc. of the European X Window System User Group Autumn Conference, 1990.

[] 7] M. Stein, "New Operational Capability for ATOP Lab," Jet Propulsion Laboratory Internal Document Interoffice Memorandum, 3474-93-008, Jan. 1993.

[18] I. E. Sutherland, "Three-Dimensional Data Input by Tablet," Proc. IEEE, vol. 62, no. 4, pp. 453-461, 1974.

[19] D. A. **Young,** *The X Window System Programming and* **Applications** *with Xt: OSF/Motif Edition*, Prentice Hall, 1990.